

S/N 10/830,164

PATENT
CONF. # 8149

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Avadhanam et al. Examiner: Khanh B. Pham

Serial No.: 10/830,164 Group Art Unit: 2166

Filed: April 21, 2004 Docket No.: MS 167378.02 / 40062.128USC1

Title: METHOD AND SYSTEM FOR CREATING A DATABASE TABLE INDEX
USING MULTIPLE PROCESSING UNITS

CERTIFICATE UNDER 37 CFR 1.8:

I hereby certify that this correspondence is being transmitted via EFS-Web to the U.S. Patent Office on April 9, 2009.

By: 

Name: Halina Wohl

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir or Madam:

Further to the Notice of Appeal mailed on January 12, 2009 for the above referenced application, Appellant submits this Brief on Appeal. A notice of Panel Decision from Pre-Appeal Brief Review was mailed January 30, 2009.

Please charge the \$510.00 fee required by 37 C.F.R. § 41.20(b)(2) to Deposit Account No. 13-2725. It is believed that no further fees are due with this Appeal Brief; however, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment with respect to this patent application to deposit account number 13-2725.

I. REAL PARTY IN INTEREST

The real party in interest is the Microsoft Corporation of Redmond, Washington.

II. RELATED APPEALS AND INTERFERENCES

A pre-appeal brief was filed in the above-identified application, Application No. 10/830,164, on January 12, 2009. The panel mailed its decision on January 30, 2009.

III. STATUS OF CLAIMS

Claims 1-26 are currently pending in the Application, but stand rejected by the Examiner. All pending claims stand rejected pursuant to the Final Office Action mailed October 10, 2008 (hereinafter "Office Action"). The Pre-Appeal Panel affirmed the Examiner's rejection in the Notice of Panel Decision from Pre-Appeal Brief Review mailed January 30, 2009 (hereinafter "Panel Decision").

Claims 1-26 are believed to be improperly rejected and are the subject of this appeal. A copy of the claims as rejected is attached as an Appendix.

IV. STATUS OF AMENDMENTS

All amendments filed during the prosecution of the above-identified application have been entered. No amendments have been filed subsequent to the Final Office Action mailed October 10, 2008.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In the following summary, Appellant has provided exemplary references to sections of the specification and drawings supporting the subject matter as defined in the claims as required by 37 C.F.R. § 41.37(c)(1)(v). The specification and drawings also include additional support for other exemplary embodiments encompassed by the claimed subject matter. Thus, it should be appreciated that the references are not exhaustive and are intended only to be illustrative in nature.

A. Independent Claim 1

Appellants note that the language of claim 1 defines the scope of the subject matter of the claim. The summary below is intended for the convenience of the Board and is not intended to limit the scope of the claim.

Claim 1 recites a method of creating an index for a database table of records using a plurality of processing units. First, partition delimiters are determined. The partition delimiters separate the database table of records into non-overlapping partitions of records. (*See* Specification, page 18, line 14-page 19, line 9, and FIG. 6, Ref. 602). Each non-overlapping partition is dedicated to a single processing unit out of the plurality of processing units for index creation. (*See id.*, page 19, lines 13-19). The partition delimiters are determined based upon a sampling of the database table of records that is used to determine a distribution of a least one key value in the records. (*See id.*, page 23, lines 2-17). After determining the partition delimiters, each processing unit accesses all of the records in the database table of records in parallel. Each processing unit filters the records in parallel to determine which records to keep. (*See id.*, page 19, lines 10-20 and FIG. 6, Ref. 604). Sub-indexes are then independently created, wherein at least two sub-indexes are created by different processors. (*See id.*, page 20, line 22 – page 21, line 5 and FIG. 6, Ref. 606). The independently created sub-indexes are then merged together into a final index related to the table. (*See id.*, page 21, lines 10-23). (*See id.*, page 21, lines 10-23 and FIG. 6, Ref. 608). The final index is then stored for later use in locating records. (*See id.*, page 17, lines 8-10).

It should be noted that support for claim 1 also lies in other portions of the written description as well as in other figures. For clarity, Appellant only cites to the above portions of the specification and figures.

B. Independent Claim 14

Appellants note that the language of claim 14 defines the scope of the subject matter of the claim. The summary below is intended for the convenience of the Board and is not intended to limit the scope of the claim.

Claim 14 recites a system for database table index creation for a database table comprising a plurality of records. A partition tool determines partition delimiters which are used

to separate the database table of records into non-overlapping partitions. (*See* Specification, page 15, lines 8-12, page 18, line 14-page 19, line 9, FIG. 4, Ref. 420, and FIG. 6, Ref. 602). The partition delimiters are determined based upon a sampling of the database table of records that is used to determine a distribution of a least one key value in the records. (*See id.*, page 15, lines 8-12, page 23, lines 2-17). Each non-overlapping partition is dedicated to a single processing unit out of the plurality of processor units for index creation. (*See id.*, page 19, lines 13-19, and FIG. 4, Ref. 420). Claim 14 additionally recites a plurality of processing units that respectively accesses all of the records in the database table in parallel. Each processing unit filters the records in parallel to determine which records to keep. (*See id.*, page. 19, lines 10-20 and FIG. 6, Ref. 604). After accessing each record and filtering out records in their assigned partition delimiter, each processor then independently creates a sub-index of the filtered records. (*See id.*, page 20, line 22 – page 21, line 5 and FIG. 6, Ref. 606). Claim 14 additionally recites a merge tool that merges the independently created sub-indexes into a final database table index. (*See id.*, page 17, lines 5-7 and FIG. 5, Ref. 502). A store tool then stores the final database table index if later use in locating records. (*See id.*, page 17, lines 8-10).

It should be noted that support for claim 14 also lies in other portions of the written description as well as in other figures. For clarity, Appellant cites only to the above portions of the specification and figures.

C. Independent Claim 18

Appellants note that the language of claim 18 defines the scope of the subject matter of the claim. The summary below is intended for the convenience of the Board and is not intended to limit the scope of the claim.

Claim 18 recites a method of creating an index using a plurality of processors. Partition delimiters are determined which are used to separate the database table of records into non-overlapping partitions of records. (*See* Specification, page 18, line 14-page 19, line 9, and FIG. 6, Ref. 602). The partition delimiters are determined based upon a sampling of the database table of records that is used to determine an approximate distribution of a least one key value in the records. (*See id.*, page 23, lines 2-17). At least one partition is dedicated to a first processing unit and at least one other partition is dedicated to a second processing unit for index creation. (*See id.*, page 19, lines 13-19). Claim 18 further recites that the first processor accesses every

record in the table of records and determines whether the individual table records are associated with the at least one partition dedicated to the first processing unit. (*See id.*, page. 19, lines 10-20 and FIG. 6, Ref. 604). The first processing unit only processes the accessed table record when the accessed table record is associated with the at least one partition dedicated to the first processing unit. (*See id.*, page 20, line 22 – page 21, line 5 and FIG. 6, Ref. 606). The result of the processing by the first processing unit is then stored for later use in locating records. (*See id.*, page 16, lines 21-23).

It should be noted that support for claim 18 also lies in other portions of the written description as well as in other figures. For clarity, Appellant cites only to the above portions of the specification and figures.

D. Independent Claim 20

Appellants note that the language of claim 20 defines the scope of the subject matter of the claim. The summary below is intended for the convenience of the Board and is not intended to limit the scope of the claim.

Claim 20 recites a method of creating an index for a database table of records using a plurality of processing units wherein more than one processing unit has access to the table. Partition delimiters are determined which are used to separate the database table of records into non-overlapping partitions. (*See* Specification, page 18, line 14-page 19, line 9, and FIG. 6, Ref. 602). The partition delimiters are determined based upon a sampling of the database table of records that is used to determine an approximate distribution of a least one key value in the records. (*See id.*, page 23, lines 2-17). Each non-overlapping partition is dedicated to one processing unit for index creation. (*See id.*, page 19, lines 13-19). Upon determining the partition delimiters, each processing unit accesses all of the records in the database table of records in parallel. Each processing unit filters the records in parallel to determine which records to keep. (*See id.*, page. 19, lines 10-20 and FIG. 6, Ref. 604). Each processor independently creates a sub-index of the filtered records. (*See id.*, page 20, line 22 – page 21, line 5 and FIG. 6, Ref. 606). Consecutive blocks of disk space are allocated for storage for each sub-index and the parts of each sub-index are stored on consecutive blocks on the disk. (*See id.*, page 30, lines 11-19 and FIG. 8, Ref. 818). The independently created sub-indexes are then merged together into a final index for the database table of records. (*See id.*, page 21, lines 10-23). (*See id.*, page 21,

lines 10-23 and FIG. 6, Ref. 608). The final index is then stored for later use in locating records in the data table of records. (*See id.*, page 17, lines 8-10).

It should be noted that support for claim 20 also lies in other portions of the written description as well as in other figures. For clarity, Appellant cites only to the above portions of the specification and figures.

E. Independent Claim 24

Appellants note that the language of claim 24 defines the scope of the subject matter of the claim. The summary below is intended for the convenience of the Board and is not intended to limit the scope of the claim.

Claim 24 recites an index creation system for creating an index of information for a table of data records. A sampling module samples the table data of records to determine sub-index delimiters. The sub-index delimiters are partition delimiters that separate that table of records into non-overlapping portions. (*See* Specification, page 15, lines 8-12, page 18, line 14-page 19, line 9, FIG. 4, Ref. 420, and FIG. 6, Ref. 602). After determining the partition delimiters, two or more index creation modules create a sub-index resulting in a plurality of sub-indexes. Each index creation module is associated with a separate processor, an access module that accesses all of the data records from the table of data records, and a filter module that filters data records according to the sub-index delimiters to keep only relevant data records. (*See id.*, page. 19, lines 10-20 and FIG. 6, Ref. 604). Each index creation module additionally comprises a sorting module that sorts the filtered data records into a sub-index. (*See id.*, page 20, line 22 – page 21, line 5 and FIG. 6, Ref. 606). The system also recites a merge module that merges sub-indexes into a final index. (*See id.*, page 17, lines 5-7 and FIG. 5, Ref. 502). Finally, a store module stores the final index for later use in locating records. (*See id.*, page 17, lines 8-10).

It should be noted that support for claim 20 also lies in other portions of the written description as well as in other figures. For clarity, Appellant cites only to the above portions of the specification and figures.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-26 stand rejected under 35 USC § 103(a) as being obvious over Gupta et al. (U.S. Patent No. 6,438,562), in view of Blank et al. (U.S. Patent No. 5,842,208), hereinafter referred to as “Gupta” and “Blank” respectively.

VII. ARGUMENT

All of the pending independent claims have been finally rejected under 35 U.S.C. § 103 as being unpatentable over Gupta in view of Blank. 35 U.S.C. § 103 states: “A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” See 35 U.S.C. § 103(a). According to the U.S. Supreme Court in *KSR International Co. v. Teleflex*, “[r]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.”¹

To establish a *prima facie* case of obviousness under 35 U.S.C. § 103(a), the references must teach or suggest all of the claimed limitations to one of ordinary skill in the art at the time the invention was made. M.P.E.P §§ 2142, 2143.03; *In re Royka*, 490 F.2d 981, 985 (C.C.P.A. 1974); *In re Wilson*, 424 F.2d 1382, 1385 (C.C.P.A. 1970). Appellants submit that the Examiner has failed to state a *prima facie* case for rejecting the independent claims, and all rejections should be withdrawn and the application allowed in its current form.

¹ *KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1741 (2007).

A. The Cited Gupta and Blank References Do Not, Alone or in Combination, Teach All of the Claimed Limitations of the Independent Claims

Claim 1

Claim 1 stands rejected as being unpatentable over Gupta in view of Blank. As discussed at length on page 11 of Appellant's November 28, 2007 Amendment (hereinafter "November Amendment"), Gupta relates to "a method, system, and product for coordinating parallel update for a global index of and indexed table." (Gupta, Abstract). "Techniques for maintaining a global index of a table during parallel data manipulations operations involve a coordinator process, data manipulation slaves and index update slaves. The coordinator process *distributes* data manipulation operations among a plurality of data manipulation slaves." (*Id.*, Col. 8, ll. 1-6) (emphasis added). Gupta teaches sorting maintenance records and determining a range by reading key values from the sorted maintenance records. (*See id.*, Col. 15, ll. 35-67). A coordinator process then uses these ranges in distributing records to multiple slave processes. (*See id.*, col. 14, ll. 9-14). The slave processes use the maintenance records distributed by the coordinator process to update a global index. (*See id.*, Col. 14, ll. 16-20).

Gupta fails to teach or suggest at least accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records. As noted above, Gupta does the opposite. First, Gupta determines a set of ranges of records by reading key values from a sorted table. After determining a set of ranges, Gupta teaches distributing the records among multiple slave processes based upon the ranges they fall into. The slave processes then perform maintenance *only* on the records which they receive. Indeed, the Office Action acknowledges that: "Gupta does not specifically teach . . . 'each processing unit accesses all of the records in the table of records.'" (Office Action, p. 4.)

Blank does not compensate for this deficiency. As discussed at length on pages 11-12 of Appellant's November Amendment, Blank relates to a "recover/build index system [that] builds an index for a file by scanning *partitions* of the file in parallel to retrieve key/rid values. The recover/build index system then sorts the scanned key/rid values for each partition in parallel." (Blank, Col. 1, ll. 37-41, emphasis added.) After the data is sorted in parallel, a "merge program merges the sort streams received from the sort programs to create a merge stream. The merge

program accepts the sort stream from two or more sort programs. Blank then performs index creation on the single data stream.

The Office Action asserts that Blank, teaches accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records, as recited in claim 1. Appellants respectfully disagree. The Office Action cites to a broad statement in the “Background of the Invention” section of Blank for support. In particular, the Office Action contends:

Applicant argued that Gupta and Blank, as combined, does not teach or suggest “each processing unit accesses all of the records in the table of records.” On the contrary, Blank teaches this limitation at Col. 1 lines 17-24 which states:

“The process involves scanning all the records in the file, extracting a key value and record identifier (rid) value from each of the records, sorting all of the key/rid values, and then building the index from the sorted key/rid value”

Office Action, p. 18.

The cited section, however, simply does not meet the limitation of accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records, as recited in claim 1. Because the recited section of Blank is part of the “Background of the Invention” section, it comprises only a short, general description of the problem Blank is attempting to solve. Specifically, Blank addresses the issue of efficiency in creating an index for a large database file: “Typically, the scanning, sorting, and index build steps are performed *serially*, which can be time consuming in the case of a large database file.” (Blank, Col. 1, ll. 22-24) (emphasis added). There is absolutely no discussion of how this system, described by Blank as prior art to the Blank patent, accomplishes “scanning all the records in the file.”

By contrast, claim 1 requires “a computing environment having a plurality of processing units . . . accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records.” This is not taught by the Background section of Blank. In fact, given that the system described in the Background section of Blank is described as operating “serially,” it is unlikely that the system accesses the table records in parallel, let alone that each processing unit accesses all of the records in the table of records.

Moreover, the actual system disclosed in the figures and detailed description of Blank also does not meet at least this limitation of claim 1. Indeed, while in Blank all of the records are scanned, it does not necessarily follow that each of the scan programs in Blank scan all of the records. Rather, Blank teaches that each scan program operates on a single partition and that, operating together in parallel, the scan programs of Blank scan all of the records. This is clearly demonstrated in Figure 1 of Blank, which shows that each partition 120 has its own scan process 108 and sort process 110. In discussing Figure 1, Blank states that “*each of the partitions 120 is scanned in parallel by the scan programs 108 . . .*” (Blank, col. 2, ll. 25-26.) Blank further teaches that each processing unit accesses only a portion of the table (i.e., each processing unit scans a *single partition*):

[t]he scan programs 108 executing in parallel extract key values (of a particular key) and record identifiers (rids) or pointers from the partitions 120 to create a key/rid or scan stream *for each partition 112*. (Blank, col. 2, l. 64 – col. 3, l. 1, emphasis added.)

The scan programs in Blank are assigned only a particular partition of the table, not all of the records in the table of records. Thus, Blank fails to teach or suggest accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records. For at least these reasons, claim 1 is allowable over the cited references.

Claim 14

For at least similar reasons, both Gupta and Blank also fail to teach or suggest all of the limitations of independent claim 14. Claim 14 recites, *inter alia*, a plurality of processing units that respectively accesses the database table in parallel, wherein each of the respective processing units accesses all of the records in the table of records. Thus, for at least similar reasons as discussed with respect to claim 1, claim 14 is allowable over the cited references.

Claim 18

For at least similar reasons as previously discussed, both Gupta and Blank also fail to teach or suggest all of the limitations of independent claim 18. Claim 18 recites, *inter alia*, wherein at least one partition is dedicated to a first processing unit for index creation and at least one other partition is dedicated a second processing unit for index creation; the first processing unit accessing every record in a table record.

Additionally, the Examiner rejected claim 18 under the same rational as claim 1, despite the fact that claim 18 recites different features. (See Office Action, pages 2-4). More specifically, claim 18 additionally recites storing a result produced by the first processing unit for later use in locating records. The result produced by the first processing unit is a sub-index. Blank teaches that the separate sort streams are immediately merged into a final index. (See FIGs. 4-5 and Col. 3, lines 10-23). Indeed, Blank makes no mention of storing a result produced by the first processing unit for later use in locating records. For at least the forgoing reasons, claim 18 is allowable over the cited references.

Claim 20

For at least similar reasons, both Gupta and Blank also fail to teach or suggest all of the limitations of independent claim 20. Claim 20 recites, *inter alia*, accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records. Thus, for at least similar reasons as discussed with respect to claim 1, claim 20 is allowable over the cited references.

Claim 24

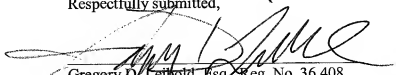
For at least similar reasons, both Gupta and Blank also fail to teach or suggest all of the limitations of independent claim 24. Claim 24 recites, *inter alia*, an access module that accesses all of the data records from the table of data records. Thus, for at least similar reasons as discussed with respect to claim 1, claim 24 is allowable over the cited references.

For at least the reasons set forth herein, Appellants respectfully request that the Board reverse the rejection of the claims and that the claims be allowed in their current form.

Date: April 9, 2009



Respectfully submitted,


Gregory D. Leibold, Esq., Reg. No. 36,408
Merchant & Gould P.C.
P.O. Box 2903
Minneapolis, MN 55402-0903
(303) 357-1642

VIII. CLAIMS APPENDIX

1. (Previously Presented) A method of creating an index for a database table of records, the method occurring in a computer environment having a plurality of processing units wherein each processing unit has access to the table, the method comprising the steps of:

determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation, wherein the step of determining comprises sampling the database table of records to determine an approximate distribution of at least one key value in the records;

accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records, wherein the step of accessing occurs after the step of determining;

filtering the accessed records in parallel, wherein each processing unit determines which records to keep;

independently creating a plurality of sub-indexes, wherein at least two sub-indexes are created by different processing units;

merging the sub-indexes together to create a final index related to the table; and

storing the final index for later use in locating records.

2. (Original) A method as defined in claim 1 wherein the act of creating the sub-indexes further comprises sorting the records and generating a data structure based on the sorted records.

3. (Original) A method as defined in claim 2 wherein the data structure is a B-Tree data structure.

4. (Original) A method a defined in claim 2 wherein the data structure has multiple levels.

5. (Original) A method as defined in claim 2 wherein the data structure is a clustered index.

6. (Original) A method as defined in claim 1 further comprising gathering sub-index statistical information and stitching sub-index statistical information.

7. (Previously Presented) A method as defined in claim 1 wherein the method is initiated by an index creation manager module.
8. (Original) A method as defined in claim 1 wherein the method is initiated by a query manager in response to a supplied query.
9. (Original) A method as defined in claim 1 wherein the method initiated automatically in response to a modification to the table.
10. (Previously Presented) A method as defined in claim 1 wherein the act of determining partition delimiters comprises:
- creating a histogram based on the sampled information; and
 - evaluation the histogram to determine the partition delimiters.
11. (Original) A method as defined in claim 10 further comprising:
- determining a processing unit goal value based on the number of processing units in the computer system;
 - determining a least common multiple value based on the processing unit goal value;
 - determining whether the histogram information may be substantially evenly split into the least common multiple value number of partitions;
 - if so, creating the partition delimiters based on the least common multiple value; and
 - if not, adjusting the processing unit goal to determine a new least common multiple value to determine partition delimiters.
12. (Previously Presented) A computer storage medium readable by a computer and encoding instructions for executing the method recited in claim 1.
13. (Previously Presented) A computer storage medium readable by a computer and encoding instructions for executing the method recited in claim 11.

14. (Previously Presented) A system for database table index creation for a database table, the database table stored in memory and comprising a plurality of records, the system comprising:

a partition tool that determines partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation, wherein the step of determining partition delimiters comprises sampling the database table of records to determine an approximate distribution of at least one key value in the records;

a plurality of processing units that respectively accesses the database table in parallel, wherein each of the respective processing units accesses all of the records in the table of records and filters the accessed records to determine which records to keep and wherein each of the respective processing units creates a sub-index of database table records resulting in a plurality of sub-indexes;

a merge tool that merges the plurality of sub-indexes into a final database table index;

and

a store tool that stores the final database table index for later use in locating records.

15. (Original) A system as defined in claim 14 wherein each processing unit further comprises:

a scanning module that scans the database table;

a filter module that filters the accessed records and selectively predetermined records;

and

a sorting module that sorts records kept by the filter module into a sub-index.

16. (Original) A system as defined in claim 15 wherein the scanning module, filter module and sorting module, for each processing unit, operates concurrently.

17. (Original) A system as defined in claim 15 further comprising a sampling module for sampling the database table and a partition module for dividing the records into substantially equal quantities related to the number of processing units.

18. (Previously Presented) A method of creating an index for a database table of records, the method occurring in a computer environment having a plurality of processing units wherein more than one processing unit has access to the table, the method comprising the steps of:

determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, wherein the step of determining comprises sampling the database table of records to determine an approximate distribution of at least one key value in the records, and wherein at least one partition is dedicated to a first processing unit for index creation and at least one other partition is dedicated a second processing unit for index creation;

the first processing unit accessing every record in a table record and determining whether the table record is associated with the at least one partition dedicated to the first processing unit;

the first processing unit only processing the accessed table record when the accessed table record is associated with the at least one partition dedicated to the first processing unit; and

storing a result produced by the first processing unit for later use in locating records.

19. (Original) A method as defined in claim 18 further comprises:

upon determining that the accessed table record is not associated with the at least one partition dedicated to the first processing unit, passing the accessed record to the second processing unit for index creation.

20. (Previously Presented) A method of creating an index for a database table of records, the method occurring in a computer environment having a plurality of processing units wherein more than one processing unit has access to the table, the method comprising:

determining partition delimiters, each partition delimiter separating the table into non-overlapping partitions of records, each partition dedicated to one processing unit for index creation, wherein the step of determining comprises sampling the database table of records to determine an approximate distribution of at least one key value in the records;

accessing the table records in parallel, wherein each processing unit accesses all of the records in the table of records, wherein the step of accessing occurs after the step of determining;

filtering the accessed records in parallel, wherein each processing unit determines which records to keep;

independently creating a plurality of sub-indexes, wherein at least two sub-indexes are created by different processing units;

allocating blocks of a disk to store each sub-index, wherein parts of each sub-index are stored on consecutive blocks on the disk;

merging the sub-indexes together to create a final index related to the table; and

storing the final index for later use in locating records.

21. (Original) A method as defined in claim 20 wherein the act of allocating portions of the disk allocates a predetermined number of blocks, the predetermined number of blocks is determined during the determination of the partition delimiters.

22. (Original) A method as defined in claim 20 wherein the allocation of portions of the disk comprises:

maintaining a cache of allocated pages and allocating pages for each partition in the cache for each processing unit; and

retrieving a pre-determined number of database pages upon request, and wherein the number of pages to allocate upon each request is determined by the size of the cache.

23. (Original) A method as defined in claim 22 wherein the cache has a size depending on the size of the index being built and the number of currently available free pages in the system.

24. (Previously Presented) In a computer system having a plurality of processing units, an index creation system for creating an index of information for a table of data records, the system comprising:

a sampling module that samples the table of data records to determine sub-index delimiters, wherein the sub-index delimiters are used as partition delimiters separating the table into non-overlapping portions of records;

two or more index creation modules, each index creation module associated with a processing unit, each index creation module creates a sub-index resulting in a plurality of sub-indexes;

a merge module that merges the sub-indexes into a final index,

wherein each index creation module comprises:

an access module that accesses all of the data records from the table of data records;

a filter module that filters data records according to the sub-index delimiters to keep only relevant data records; and

a sorting module that sorts the relevant data records into a sub-index; and

a store module that stores the final index for later use in locating records.

25. (Original) A system as defined in claim 24 further comprising a memory allocation module that allocates parts of memory for storing the sub-indexes, and wherein the memory allocation module allocates a predetermined number of parts, the predetermined number of parts is determined during the determination of the delimiters.

26. (Original) A system as defined in claim 24 further comprising a cache memory module that manages a cache of allocation pages and allocates pages for storing each sub-index in the cache and wherein the number of pages allocated to the cache is determined upon determining the delimiters.

27-28. (Canceled)

IX. EVIDENCE APPENDIX

Office Actions and Amendments/Responses

1. Final Office Action mailed June 28, 2007
2. Response to Final Office Action filed November 28, 2007
3. Supplemental Amendment filed January 15, 2008
4. Office Action mailed March 19, 2008
5. Amendment Mailed June 19, 2008
6. Final Office Action mailed October 10, 2008
7. Pre-Appeal Brief Request for Review Supporting Statement filed January 12, 2009

References Relied Upon by the Examiner:

1. Gupta et al., U.S. Patent No. 6,438,562
2. Blank et al., U.S. Patent No. 5,842,208

X. RELATED PROCEEDINGS APPENDIX

A copy of the Notice of Panel Decision from Pre-Appeal Brief Review has been included herewith.